

Sleepiest Guard Challenge

Assumptions:

- The file is accessible.
- The format of each line is valid. In addition, there is no missing data, meaning – if we have information regarding a certain date we have all the information – a line informing on a guard starting his shift, and if a guard were to fall asleep then a line informing on his falling asleep and a line informing on his waking up.
- Guards show up to work awake – meaning, when the shift starts, which is at 00:00 (or after), they are not yet asleep (so we don't have to worry about them getting early).
- Guards count as asleep the same minute they fall asleep, and they count as awake the same minute they wake up. For example: if a guard fell asleep at 00:05 and woke up at 00:09 then he is considered asleep at 00:05, 00:06, 00:07 and 00:08, and he is considered awake at 00:09.

Implementation Idea:

- In order to look at the data in a chronological order (as in the provided example), I used the built-in sort function.

Parsing:

We have three sorts of input lines:

- a. A line informing of a guard starting a shift.
- b. A line informing of that guard falling asleep.
- c. A line informing of that guard waking up.

I used a **dictionary** where the **keys** are the **guard ids** (int) and the **values** are the **lists of minutes they were asleep** in (list of int).

While going over the sorted file, for each line, I checked what sort of line it is (a/b/c):

Case a: I extracted the guard id using a regular expression.

Case b: I extracted the time the guard fell asleep in using a regular expression.

Case c: I extracted the time the guard woke up in using a regular expression.

Then, using python's range function, I created a list of all the minutes during which this guard was asleep and added this list to the dictionary accordingly.

- Then, to get the guard that slept the most I calculated the one that has the **longest list** of minutes he slept in.
- Then, I used hashing to get the minute that guard slept in the most:

I created a **dictionary** where the **keys** are the **minutes** and the **values** are **counters** (how many times did that guard sleep in that minute), where I went over the list of minutes the guard slept in and filled in my dictionary accordingly.

Afterwards, I found the **max** value.

- The output is printed accordingly (for example: "Guard #77 is most likely to be asleep in 00:33") and a tuple with the guard id and the minute he slept in the most is returned.

Testing:

For testing, I used both the file provided in the assignment instructions and another longer, unordered file.

I used the result tuple from the function and essentially verified each part of the tuple separately.

In each case, I first checked that the guard id is as expected, and then checked that the returned minute is correct.

In both cases, an informative message is printed.

I avoided running my tester in order not to "junk" with too many prints (it is "commented").